IN THE SPECIFICATION

Paragraph [0021] is sought to be amended as follows:

[0021]   The second application 118 is a messaging application which enables messages received from a client process residing on a first computer platform, for example, the computer platform 104-2, to be directed to a server process residing on a second computer platform, here, the server process 106 residing on the computer platform 102. The messaging application 118 further enables replies from the server process residing on the second computer platform, here, the server process 106 residing on the second computer platform 102, to be returned to the first computer platform, for example, the computer platform ~~108-2~~ 104-2. As will be more fully described below, the messaging application 118 manages a portion of the memory subsystem 112, hereafter referred to as messaging application database 140. Within the messaging application database 140, the messaging application ~~120~~ 118 maintains plural message queues as well as control information which defines, among other items, the attributes of each queue maintained thereby. Commercially available software suitable for use as the messaging application 118 is IBM's MQ Series messaging software.

10

Paragraph [0025] is sought to be amended as follows:

[0025]   A queue-based messaging system is comprised of a messaging application which

handles messages for a server process, for example, messaging application ~~110~~ **118**, and a

messaging application which handles messages for a client process, for example, one or more of

the messaging applications 112-1 through 112-X. Referring next to Figure 2, the messaging

application ~~110~~ **118** residing on the local computer platform 102 which handles messages for the

server process application 106 and the messaging application which handles messages for a client

process application on a remote computer platform, for example, the messaging application 112-

2 which handles messages for the client process application 108-2 residing on the remote

computer platform 104-2 will now be described in greater detail. As may now be seen, a queue

manager 126 resides on the messaging application ~~110~~ **118** and a queue manager 128 resides on

the messaging application 112-2. The queue managers 126 and 128 are system programs that

provide queuing services to applications. For example, the queue manager 126 provides queuing

services to the process application 106 while the queue manager 128 provides queuing services

to the process application 108-2. The queue managers 126 and 128 further provide application

programming interfaces so that the respective application processes 106 and 108-2 serviced

thereby can put messages on, and get messages from, queues. Finally, the queue managers 126

and 128 provide additional functions so that administrators can create new queues, alter the

properties of existing queues, and control the operation of the queue manager itself. It should be

noted that, while Figure 2 shows a single queue manager 126 and 128 running on the messaging

applications ~~110~~ **118** and 112-2, respectively, it should be clearly understood that plural queue

managers may run on a single messaging application. For example, for the aforementioned MQ

Series messaging software, plural queue managers may run on MVS, OS/2, Windows NT,

11

. Digital OpenVMS and UNIX systems. It should be further noted that while Figure 2 shows a single process application 106 making use of the services of the queue manager 126, plural applications, whether related to one another or entirely unrelated to one another, may make use of the queue manager 126 at the same time.

12

30760.01/4000.05900

Paragraph [0026] is sought to be amended as follows:

[0026]   Figure 2 also shows plural queues residing on each of the messaging applications 110 118 and 112-2. In this regard, it should be noted that queues reside in, and are managed by, a queue manager and have been shown in Figure 2 as being discrete relative to the queue manager merely for ease of illustration. Before using a queue, it must be open for a specified purpose. The purposes for which a queue may be opened include:

- Browsing messages on a queue;
- Retrieving messages from the queue;
- Putting messages on the queue;
- Inquiring about attributes of the queue; and
- Setting the attributes of the queue.

13

Paragraph [0028] is sought to be amended as follows:

[0028]   Residing in the queue manager 126 are a transmission queue 130, an alias queue 132, a plurality of local queues 134-1 through 134-M and a plurality of remote queues 136-1 through 136-N. Similarly, residing in the queue manager 128 are a transmission queue ~~138~~ 139, a remote queue 140 and a local queue 142. The transmission queues ~~132~~ 130 and 139 are local queues which hold messages destined for a remote queue. The messages held in the transmission queue are forwarded to their destination queue by the queue manager when a channel is available. The local queues 134-1 through 134-M and 142 are trigger-initiated local queues. When the queue manager places a message on a trigger-initiated local queue and certain conditions are met on that queue, a trigger message is automatically sent to the server process application 106 to notify the server process application 106 that a message has been placed on that queue. In turn, the server process application 106 will then retrieve the message from the trigger-initiated local queue. A condition commonly used for trigger-initiated local queues is that a trigger message will be issued to notify the server process application 106 that a message has been placed on the trigger-initiated local queue if there are no messages sitting in that queue when the message is placed in the queue. Finally, the queues 136-1 through 136-N and 140 are remote queues or, more precisely, local definitions of a remote queue where a message, when placed in the remote queue, is to be delivered. For example, the remote queue 136-2 holds the information necessary to deliver a message to the local queue 142. Conversely, the remote queue 140 holds the information necessary to deliver a message to the local queue 134-2.

14

Paragraph [0031] is sought to be amended as follows:

[0031]   Continuing to refer to Figure 2, the process by which a message is delivered from a client process, specifically, the process application 108-2, to a server process, specifically, the process application 106, will now be described in greater detail. The client process application 108-2 will first connect to the queue manager 128 so that the client process application 108-2 may request that the queue manager 128 open the remote queue 140 which contains a description of the destination of the message. Once the remote queue 140 is opened, the client process application 108-2 places the message in the remote queue 140. As previously set forth, the remote queue 140 contains that information necessary to direct the message to the local queue 134-2 where it may be retrieved by the server process application 106. The queue manager 128 then places the message onto the transmission queue ~~138~~ <u>139</u> where it will be transmitted over the channel 105-2 to the transmission queue 130. Upon arriving at the transmission queue 130, the queue manager 126 will assume control of the message. From the transmission queue 130, the queue manager 126 will first direct the message to the alias queue 132 (if, as here, plural initiation queues are accessed through a single alias queue) and on to the trigger-initiated local queue 136-2. If there are no messages sitting in the trigger-initiated local queue 136-2 upon arrival of the message, the queue manager 126 will notify the server process application 106 of the arrival of the message. The server process application 106 will then connect to the queue manager 126 to request that the queue manager 126 open the trigger-initiated local queue 136-2 and, after the trigger-initiated local queue 136-2 has been opened, the server process application 106 retrieves the message from the trigger-initiated local queue 136-2.

15

Paragraph [0094] is sought to be amended as follows:

[0094]   Returning momentarily to Figure 1, it is contemplated that the monitoring tool 120 may contain program code which includes a first set of instructions to enable the network administrator to interact exclusively with the monitoring tool 120, for example, in an interactive dialogue, to perform the steps set forth in the method of Figure 3 to select the queues to be monitored, to select the attributes of the selected queues for which the current values are to be acquired and to initiate any appropriate corrective action. It is further contemplated that the monitoring tool 120 may contain additional program code which includes a second set of instructions to enable the monitoring tool 120 to instruct the messaging API 138 of the messaging application ~~110~~ 118 to retrieve the selected attributed values for the monitoring tool 120 and/or initiate any selected corrective action. Finally, it is still further contemplated that the monitoring tool 120 may contain further additional program code which includes a third set of instructions to enable the monitoring tool 120 to construct the network monitoring table 250 from the list of queues and attributes selected by the network administrator and the values of the selected attributes provided by the messaging API 138 and to generate a display of the network monitoring table 250 at the user interface 122. Of course, it should be understood that the network administrator could access the messaging API 138 through the user interface 122 to perform a number of these operations. However, a number of advantages are derived by having the monitoring tool 120 access the messaging API 138 on the network administrator's behalf. For example, the network administrator is protected from inadvertently modifying the queues and/or their attributes when acquiring the current values of selected attributes. Furthermore, the queue-based messaging system can be monitored without the network administrator having detailed knowledge of the messaging API 138 itself.

16

.10760.01/4000.05900

PAGE 17/25 * RCVD AT 1/9/2006 5:44:07 PM [Eastern Standard Time] * SVR:USPTO-EFXRF-6/25 * DNIS:2738300 * CSID:9727312289 * DURATION (mm-ss):05-44